ECS Process Framework

Presentation Overview

1. What is ECS Process Framework?
2. What is it's purpose?
3. What are it's capabilities?
4. What is it's basic design?

ECS Process Framework

What is it?

It is an extensible mechanism for ECS client and server applications to transparently include ECS infrastructure features.

- **To be used solely by ECS custom developed applications**
- **Not meant for COTS applications**

ECS Process Framework

What is its purpose?

- Encapsulates implementation details of ECS infrastructure services and removes the need for programmers to rewrite common initialization code.
- Ensures design and implementation consistency for all ECS Client and Server Applications.
- Provides a basis by which future extensions to infrastructure mechanisms can be incorporated without adversely affecting the ECS developers.

ECS Process Framework

What are its capabilities?

(1) Ability to initialize the process application and infrastructure in a consistent way and provide some basic process information

(2) Interface to Mode Management

(3) Interface to Error-Event Logging

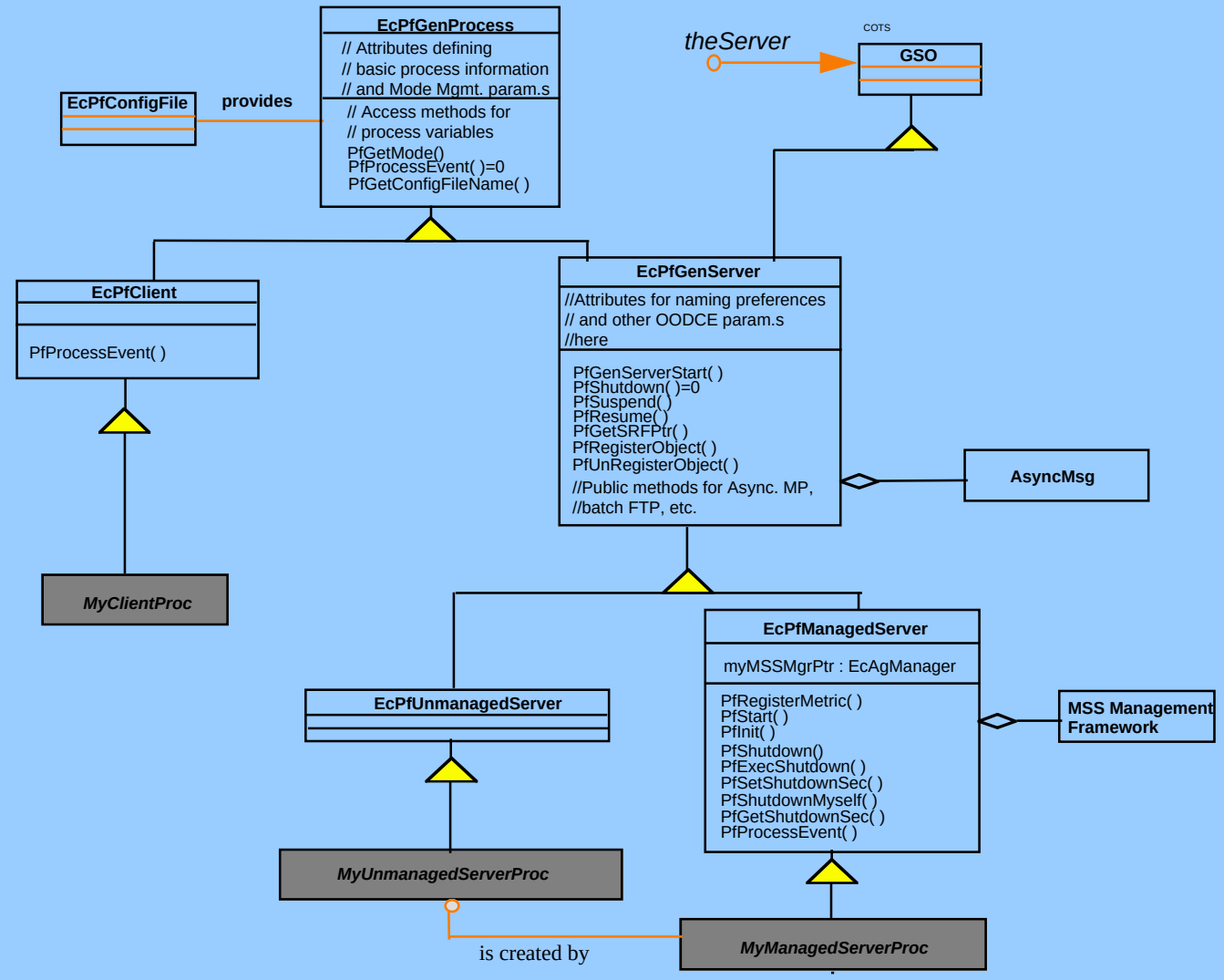
(4) Ability to set Naming/Directory Service

(5) Ability to set Security management parameters

(6) Support to Life Cycle management

(7) Interface to Asynchronous Messaging management

PF Software Design
Simplified Object Model



Configuration file

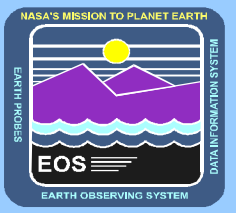
- Server process options are indicated in an orderly (parsable) fashion in a configuration file - like the .Xdefaults in X Windows
- This file *can* be different for each instantiation of a server executable
- Code should not need to be recompiled to run with different options (ex: different modes)
- Several standard options have been identified
- Other options as identified by subsystems will be incorporated.

Server Request Framework

What is it?

- Framework for constructing ECS Servers & Clients APIs
- Encapsulates communication technology easing transition to other technologies (e.g. CORBA)

Server Request Framework



Purpose?

- Hide details of communications mechanism
- Allows server developers to concentrate on services, not infrastructure
- Provide single common implementation of asynchronous request processing requirements
- Provide a simple interface for server developers

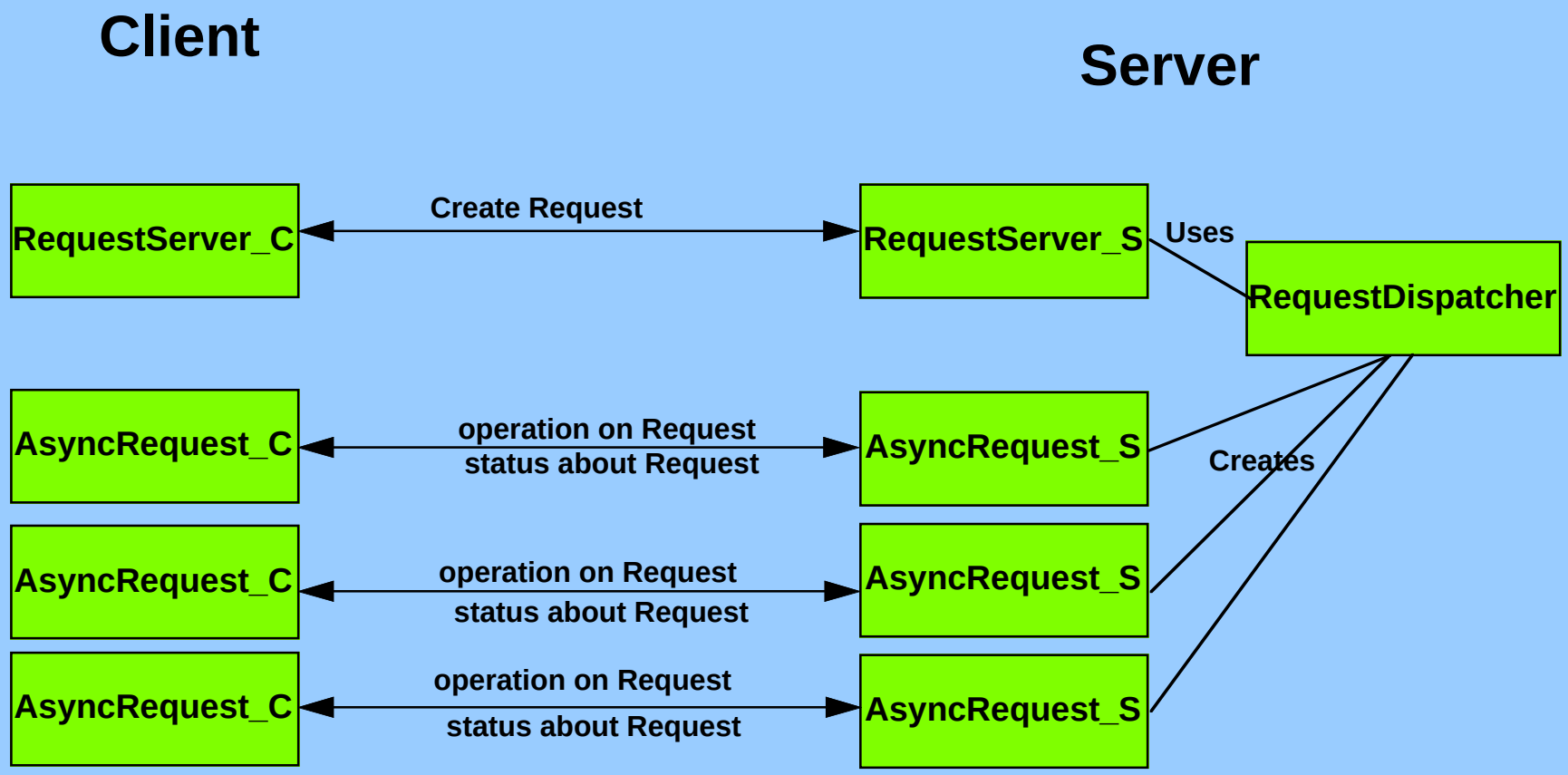
Server Request Framework

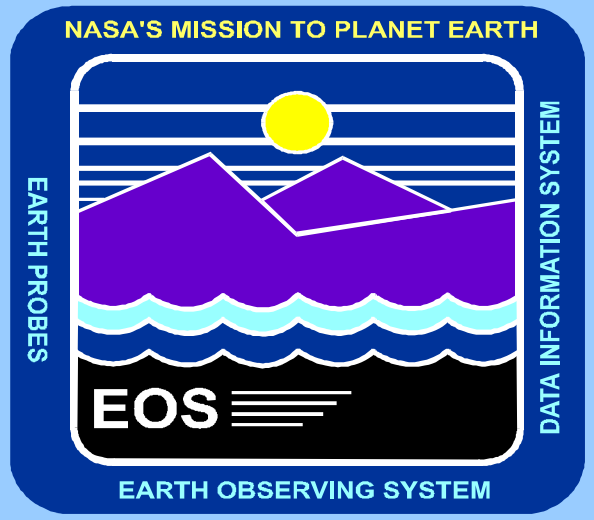
Capabilities?

- Encapsulates common requirements for asynchronous request processing
- For persistence

Server Request Framework

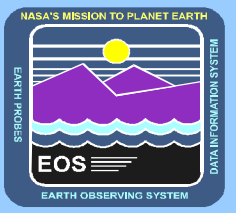
Logical View





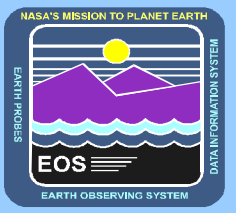
Event Handling

Road Map



- Definition of Events and Errors
- Event Classifications
- Event Resource Catalog
- Event Attribute Assignment
- Event Usage
- Error Logging APIs
- Event Handling for COTS

Errors and Events



- **Events**
  - **Broad category that encompasses normal occurrences and abnormal conditions**
  - **management events (MSS), e.g., faults, performance, etc.**
  - **non-management events (non-MSS), e.g. debug messages, etc.**
- **Errors**
  - **unexpected problem**
  - **deviation from the correct result**
  - **e.g. “tape/disk read/write error”**

Event Classifications

- **Event Category**
  - **Highest level of categorization of an event. For MSS filtering (HPOV)**
- **Event Type**
  - **Lower level of event classification. Must be grouped with an Event Category. Used for fault correlation (Tivoli).**
- **Event Priority**
  - **Event level or severity. Used for Agent filtering.**
- **Event Subsystem**
  - **Specifies the subsystem that the event was created for.**
- **Event CSCI (EcTAgCsci)**
  - **Specifies the CSCI that the event was created in.**

Event Classifications (cont)

- Log Type (specified by application programmer in call to PfProcessEvent)
  - Management Log - Events of significance to Operations. Logged to Local MSS Log and depending on severity level, sent realtime to MSS for operator notification. (EcEAgMssLog)
  - Application Log - Events of no significance to Operations. Logged to Local Applications log. (default) (EcEAgAppLog)
- All Event Classifications are enum types defined in EcAgCommon.h

Event Category

- **Event Category Classifications (EcTAgEventCategory) - Provided in the Resource Catalog**
  - **Fault**
    - **high priority error events that have an impact on operations**
    - **caused by an error which was detected by the sending application**
  - **Performance**
    - **Generated when the measurement of performance metrics exceeds the thresholds**

Event Category (cont.)

- **Event Category Classifications (EcTAgEventCategory) - Provided in the Resource Catalog**
  - **SecurityAlert**
    - **Generated when a security violation has been detected**
  - **TopologyChange**
    - **Generated when a change of state in a managed resource is detected**
  - **Transaction**
    - **Generated for order/request tracking purposes. Not an error or warning.**

Event Type

Event Type (EcTAgEventType) - Provided in the resource catalog.

- FaultDCE
- FaultOODCE
- FaultSybase
- FaultUnix
- FaultAppSpecific
- TransactionOrderStart
- TransactionOrderStop
- TransactionRequestStart
- TransactionRequestStop
- SecurityAccessViolation
- SecurityLoginFailure
- PerformanceThreshold
- PerformanceAppSpecific
- PerformancePoll
- TopologyStarting
- TopologyStarted
- TopologyStopping
- TopologyStopped
- TopologyDied
- TopologyInstalled
- TopologyUninstalled

All Topology Events are MSS events. They are not applicable to non-MSS software developers.

Event Priority

- Event Priority (EcTAgPriority) - Provided in the Resource Catalog
  - Level 0 - Low
    - Events that are useful to record but do not have any adverse significance in the operational status of the system.
      - i.e. login successful
  - Level 1 - Medium- Low
    - Warnings or errors of low significance which are recoverable.
      - i.e. memory/stack overflow errors that are recoverable
  - Level 2 - Medium
    - Events that are potential indicators of future problems; the system is fully operational.
      - i.e. performance threshold exceeded, PGE terminated abnormally

  Level 3 - High
    - Error events which have an impact on operations.
      - i.e. Autosys process(es) died, V0 Gateway down

Event Subsystem

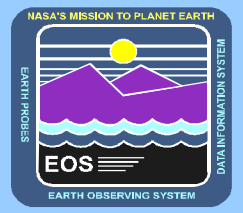
- Event Subsystem (EcTAgSubsys)
  - EcEAgSsCss Communication Subsystem
  - EcEAgSsDms Data Management Subsystem
  - EcEAgSsDps Data Processing Subsystem
  - EcEAgSsDss Data Server Subsystem
  - EcEAgSsIns Ingest Subsystem
  - EcEAgSsIos Interoperability Subsystem
  - EcEAgSsIss Internetworking Subsystem
  - EcEAgSsMss Systems Management Subsystem
  - EcEAgSsPdps Planning and Data Processing Subsystem

Event CSCI (Examples)

Event CSCI (EcTAgCsci) - subclassification of Event Subsystem

EcEAgCsIoAd	ADSRV
EcEAgCsDpAt	AITTL
EcEAgCsCm	Common SW
EcEAgCsCsDc	DCI
EcEAgCsDmDd	DDICT
EcEAgCsDsDd	DDIST
EcEAgCsDsDo	DDSRV
EcEAgCsClDt	DESKT
EcEAgCsDmDm	DIMGR
EcEAgCsDmGw	GTWAY
EcEAgCsIt	INCI
EcEAgCsIn	INGST
EcEAgCsDmLm	LIMGR
EcEAgCsMsAg	MACI
EcEAgCsMsMc	MCI
EcEAgCsMsMl	MLCI
EcEAgCsPl	PLANG
EcEAgCsDpPr	PRONG
EcEAgCsDpTk	SDPTK
EcEAgCsDsSd	SDSRV
EcEAgCsDsSt	STMGT
EcEAgCsClWb	WKBCH

Resource Catalog



- Application developers must provide a file defining information about each event reported by their code.
- This file is in binary format.
- The Event Resource Catalog is external to the application.
- The error/event handling infrastructure will insert the associated information from the resource catalog into the error/event object with it is instantiated.

Summary

- **CSS Supports:**
  - **Universal Refrences (UR)**
  - **Process Framework (PF)**
  - **Server Request Framework (SRF)**
  - **Error/Event Logging**
  - **Subscription Service**